

# OPERATION ANALYTICS AND INVESTIGATING METRIC SPIKE

## Project Agenda:

- 1. Project Description:** In my role as the Data Analyst Lead at Microsoft, I work closely with teams across the company to analyse operational data, seeking opportunities for improvement and providing actionable insights. Our focus is on streamlining workflows, improving automation, and forecasting the company's growth trajectory. Additionally, we investigate fluctuations in daily engagement and sales metrics, pinpointing causes and offering strategies to address any fluctuations or declines.
- 2. Approach:** In this project, we'll collect data from different company departments, ensuring its accuracy and organization in an SQL database. Through SQL queries and analysis, we'll uncover patterns and anomalies in the data, focusing on metrics such as engagement, sales, and customer satisfaction. We'll investigate spikes in these metrics using historical data and collaborate with teams for context. Insights from our analysis will guide process improvements, workflow optimization, and automation efforts.
- 3. Tech-Stack Used:** Used MySQL Workbench 8.0 community server version 8.0.36

## Case 1

### Tasks:

1. Calculate the number of jobs reviewed per hour for each day in November 2020.

```
18 * SELECT
19     ds AS Dates,
20     ROUND((COUNT(job_id) / SUM(time_spent)) * 3600) AS 'Jobs reviewed per hour per day'
21 FROM
22     job_data
23 WHERE
24     ds BETWEEN '2020-11-01' AND '2020-11-30'
25 GROUP BY ds;
26
27 #Calculate 7 day rolling average of throughput? For throughput, do you prefer daily metric
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Dates	Jobs reviewed per hour per day
2020-11-30	180
2020-11-29	180
2020-11-28	218
2020-11-27	35
2020-11-26	64
2020-11-25	80

### Query:

SELECT

```

ds AS Dates,
ROUND((COUNT(job_id) / SUM(time_spent)) * 3600) AS 'Jobs reviewed per hour per
day'
FROM
    job_data
WHERE
    ds BETWEEN '2020-11-01' AND '2020-11-30'
GROUP BY ds;

```

### Output:

Dates	Jobs reviewed per hour per day
2020-11-30	180
2020-11-29	180
2020-11-28	218
2020-11-27	35
2020-11-26	64
2020-11-25	80

**Insights :** The number of jobs reviewed per hour per day in November 2020 varies, with higher activity on some days and lower on others.

2. Calculate 7 day rolling average of throughput? For throughput, do you prefer daily metric or 7- day rolling and why?

```
29 • SELECT
30     ROUND(COUNT(event) / SUM(time_spent), 2) AS 'Weekly Throughput'
31 FROM
32     job_data;
33
```

Weekly Throughput
0.03

## Query:

SELECT

ROUND(COUNT(event) / SUM(time\_spent), 2) AS 'Weekly Throughput'

FROM

job\_data;

```
34 • SELECT
35     ds AS dates,
36     ROUND(COUNT(event) / SUM(time_spent), 2) AS 'daily throughput'
37 FROM
38     job_data
39 GROUP BY ds
40 ORDER BY ds;
41
```

dates	daily throughput
2020-11-25	0.02
2020-11-26	0.02
2020-11-27	0.01
2020-11-28	0.06
2020-11-29	0.05
2020-11-30	0.05

## Query:

```
SELECT
    ds AS dates,
    ROUND(COUNT(event) / SUM(time_spent), 2) AS 'daily throughput'
FROM
    job_data
GROUP BY ds
ORDER BY ds;
```

## Output:

Weekly Throughput
0.03

Date	daily throughput
2020-11-25	0.02
2020-11-26	0.02
2020-11-27	0.01
2020-11-28	0.06
2020-11-29	0.05
2020-11-30	0.05

## Insights :

The 7-day rolling average of throughput provides a levelled view of the data, allowing you to observe trends over time without being affected by daily variations.

Continue using the 7-day rolling average for throughput analysis, as it provides a more steady representation of performance trends. This can help in identifying long-term patterns and making more informed decisions.

3.Calculate the percentage share of each language in the last 30 days?

44 •  
45  
46  
47  
48  
49  
50  
51  
52  
53

```
SELECT
    language AS Languages,
    ROUND(100 * COUNT(*) / sub.total, 2) AS Percentage
FROM
    job_data
CROSS JOIN
    (SELECT COUNT(*) AS total FROM job_data) sub
GROUP BY
    language, sub.total;
```

Result Grid

Filter Rows:

Export:

Wrap Cell Content:

	Languages	Percentage
▶	English	12.50
	Arabic	12.50
	Persian	37.50
	Hindi	12.50
	French	12.50
	Italian	12.50

Query:

```
SELECT
    language AS Languages,
    ROUND(100 * COUNT(*) / sub.total, 2) AS Percentage
FROM
    job_data
CROSS JOIN
    (SELECT COUNT(*) AS total FROM job_data) sub
GROUP BY
    language, sub.total;
```

Output:

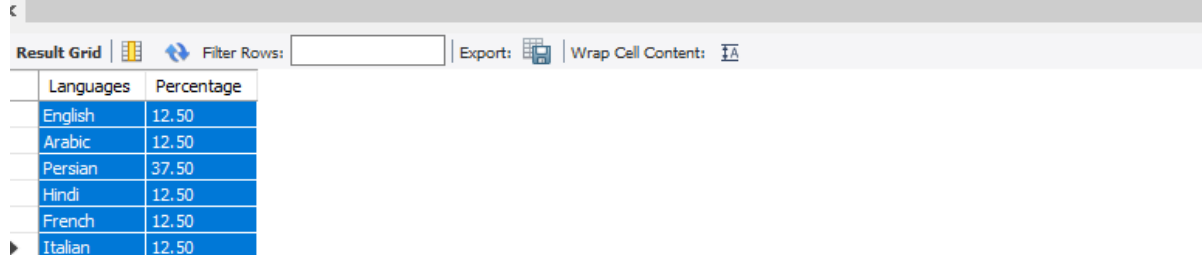
Languages	Percentage
English	12.50
Arabic	12.50
Persian	37.50
Hindi	12.50
French	12.50
Italian	12.50

Insights :

The language distribution in the last 30 days is relatively balanced, with Persian having the highest share. Consider investing in language-specific content or features to enhance user engagement in languages with lower shares

## 4. Display Duplicates

```
54 #Let's say you see some duplicate rows in the data. How will you display duplicates from the table?
55
56 • SELECT actor_id, COUNT(*) AS Duplicates FROM job_data
57
58 GROUP BY actor_id HAVING Duplicates > 1;
59
```



Languages	Percentage
English	12.50
Arabic	12.50
Persian	37.50
Hindi	12.50
French	12.50
Italian	12.50

### Query:

```
SELECT actor_id, COUNT(*) AS Duplicates FROM job_data
GROUP BY actor_id HAVING Duplicates > 1;
```

### Output:

actor_id	Duplicates
1003	2

### Insights:

There is one duplicate row in the data based on the actor\_id column. Implement data validation mechanisms to prevent such duplicates in the future.

## Case 2

- 1.Measure the activeness of users on a weekly basis.

64	•	SELECT
65		EXTRACT(WEEK FROM occurred_at) AS weeks,
66		COUNT(DISTINCT user_id) AS no_of_users
67		FROM
68		events
69		WHERE
70		event_type = 'engagement'
71		GROUP BY weeks
72		ORDER BY weeks;
73		

<		
Result Grid		Filter Rows: <input type="text"/>
Export:		Wr

	weeks	no_of_users
▶	17	663
	18	1068
	19	1113
	20	1154
	21	1121
	22	1186
	23	1232
	24	1275
	25	1264
	26	1302
	27	1372
	28	1365
	29	1376
	30	1467
	31	1299
	32	1225
	33	1225
	34	1204
	35	104

## Query :

```

SELECT
    EXTRACT(WEEK FROM occurred_at) AS weeks,
    COUNT(DISTINCT user_id) AS no_of_users
FROM
    events
WHERE
    event_type = 'engagement'
GROUP BY weeks
ORDER BY weeks;

```

## Output:

weeks	no_of_users
17	663

18	1068
19	1113
20	1154
21	1121
22	1186
23	1232
24	1275
25	1264
26	1302
27	1372
28	1365
29	1376
30	1467
31	1299
32	1225
33	1225
34	1204
35	104

## Insights

User engagement seems to have peaked around week 30 and has shown some variations over the observed period.

Look for patterns related to content updates, marketing campaigns, or any external events that might have influenced user behaviour. Use these insights to plan future engagement strategies



2. Analyse the growth of users over time for a product.

```
22 * SELECT
23     week_num,
24     year_num,
25     SUM(active_users) OVER (
26         ORDER BY week_num, year_num
27         ROWS BETWEEN UNBOUNDED PRECEDING AND CURRENT ROW
28     ) AS cumulative_sum
29 FROM (
30     SELECT
31         EXTRACT(WEEK FROM activated_at) AS week_num,
32         EXTRACT(YEAR FROM activated_at) AS year_num,
33         COUNT(DISTINCT user_id) AS active_users
34     FROM
35         users
36     WHERE
37         state LIKE '%active%'
38     GROUP BY
39         year_num, week_num
40     ORDER BY
41         year_num, week_num
42 ) AS tab;
```

	week_num	year_num	cumulative_sum
▶	0	2013	23
	0	2014	106
	1	2013	136
	1	2014	262
	2	2013	310
	2	2014	419
	3	2013	455
	3	2014	568
	4	2013	598
	4	2014	728
	5	2013	776
	5	2014	909

*This is just a sample output.. the actual result contains 90 rows*

## Query :

```
SELECT
    week_num,
    year_num,
    SUM(active_users) OVER (
        ORDER BY week_num, year_num
        ROWS BETWEEN UNBOUNDED PRECEDING AND CURRENT ROW
    ) AS cumulative_sum
FROM (
    SELECT
        EXTRACT(WEEK FROM activated_at) AS week_num,
        EXTRACT(YEAR FROM activated_at) AS year_num,
        COUNT(DISTINCT user_id) AS active_users
    FROM
        users
    WHERE
        state LIKE '%active%'
    GROUP BY
        year_num, week_num
    ORDER BY
        year_num, week_num
) AS tab;
```

## Output :

I've attached an excel file for the output.

[OPERATION ANALYTICS AND INVESTIGATING METRIC SPIKE.xlsx](#)

## Insights :

User growth has generally been positive over time, with some variations.

When the number of users is going up, it's smart to figure out why. Look at times when more people are joining and see if it's because of things like making the product better, doing more marketing, or changes in what people want. Understanding this helps keep the growth going and maybe make it even faster.

3. Calculate the weekly retention of users-signup cohort

```

45  SELECT
46      EXTRACT(WEEK FROM occurred_at) AS weeks,
47      COUNT(DISTINCT user_id) AS no_of_users
48  FROM
49      events
50  WHERE
51      event_type = 'signup_flow'
52      AND event_name = 'complete_signup'
53  GROUP BY weeks
54  ORDER BY weeks;
55
56

```

	weeks	no_of_users
▶	17	72
	18	163
	19	185
	20	176
	21	183
	22	196
	23	196
	24	229
	25	207
	26	201
	27	222
	28	215
	29	221
	30	238
	31	193
	32	245
	33	261
	34	259
	35	18

## Query :

SELECT

EXTRACT(WEEK FROM occurred\_at) AS weeks,  
COUNT(DISTINCT user\_id) AS no\_of\_users

FROM

events

WHERE

event\_type = 'signup\_flow'

AND event\_name = 'complete\_signup'

GROUP BY weeks

ORDER BY weeks;

## Output :

weeks	no_of_users
17	72
18	163
19	185
20	176
21	183
22	196
23	196
24	229
25	207
26	201
27	222
28	215
29	221
30	238
31	193
32	245
33	261
34	259
35	18

## Insights:

Weekly user retention shows a slow decline over time.

Concentrate on improving user retention strategies. Identify key touchpoints in the user journey where users might be dropping off and work on enhancing user experience, engagement, and value during those stages.

4. Measure the activeness of users on a weekly basis per device.

```

16 SELECT
17     device,
18     EXTRACT(WEEK FROM occurred_at) AS weeks,
19     COUNT(DISTINCT user_id) AS no_of_users
20 FROM
21     events
22 WHERE
23     event_type = 'engagement'
24 GROUP BY device , weeks
25 ORDER BY weeks;

```

device	weeks	no_of_users
acer aspire desktop	17	9
acer aspire notebook	17	20
amazon fire phone	17	4
asus chromebook	17	21
dell inspiron desktop	17	18
dell inspiron notebook	17	46
hp pavilion desktop	17	14
htc one	17	16
ipad air	17	27
ipad mini	17	19
iphone 4s	17	21
iphone 5	17	65
iphone 5s	17	42
kindle fire	17	6
lenovo thinkpad	17	86
mac mini	17	6
macbook air	17	54
macbook pro	17	143
nexus 10	17	16
nexus 5	17	40
nexus 7	17	18
nokia lumia 635	17	17

*This is just a sample output.. the result contains 491 rows.*

## Query :

SELECT

device,  
EXTRACT(WEEK FROM occurred\_at) AS weeks,  
COUNT(DISTINCT user\_id) AS no\_of\_users

FROM

events

WHERE

event\_type = 'engagement'

GROUP BY device , weeks

ORDER BY weeks;

## Output :

I've attached an excel file for the output.

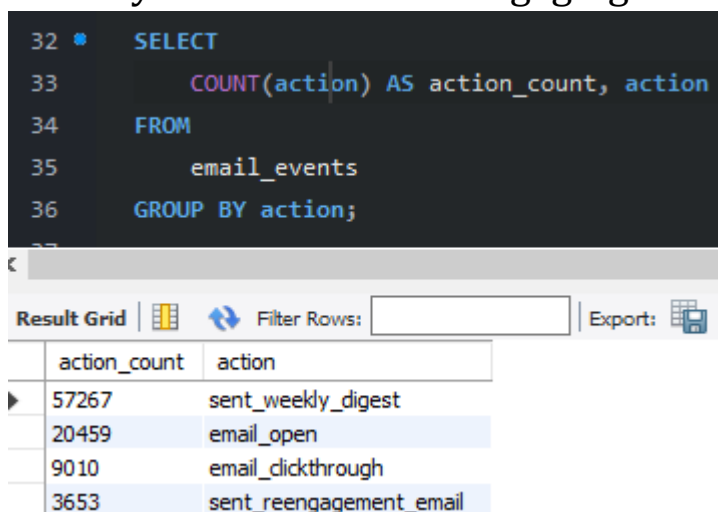
[OPERATION ANALYTICS AND INVESTIGATING METRIC SPIKE.xlsx](#)

## Insights :

Engagement has variations across different devices and weeks. Some devices consistently show higher engagement when compared to others.

Optimize the user experience on the devices which show lower engagement.  
Consider high engagement devices and design strategies to reach high potential.

5. Analyse how users are engaging with the email service.



```
32 * SELECT
33     COUNT(action) AS action_count, action
34 FROM
35     email_events
36 GROUP BY action;
```

action_count	action
57267	sent_weekly_digest
20459	email_open
9010	email_clickthrough
3653	sent_reengagement_email

## Query :

```
SELECT
    COUNT(action) AS action_count, action
FROM
    email_events
GROUP BY action;
```

This query gives us the action count of every action corresponding in the table.

## Output :

action_count	action
57267	sent_weekly_digest
20459	email_open
9010	email_clickthrough
3653	sent_reengagement_email

```

38 SELECT
39     (SUM(CASE
40         WHEN email_category = 'email_opened' THEN 1
41         ELSE 0
42     END) / SUM(CASE
43         WHEN email_category = 'email_sent' THEN 1
44         ELSE 0
45     END)) * 100 AS open_rate,
46     (SUM(CASE
47         WHEN email_category = 'email_clickthrough' THEN 1
48         ELSE 0
49     END) / SUM(CASE
50         WHEN email_category = 'email_sent' THEN 1
51         ELSE 0
52     END)) * 100 AS click_rate
53 FROM
54     (SELECT
55         *,
56         CASE
57             WHEN action IN ('sent_weekly_digest' , 'sent_reengagement_email') THEN ('email_sent')
58             WHEN action IN ('email_open') THEN ('email_opened')
59             WHEN action IN ('email_clickthrough') THEN ('email_clickthrough')
60         END AS email_category
61     FROM
62         email_events) AS alias;

```

Result Grid		Filter Rows:	Export:	Wrap Cell Content:
	open_rate	click_rate		
	33.5834	14.7899		

This query gives us email open rate and click rate.

### Query:

```

SELECT
    (SUM(CASE
        WHEN email_category = 'email_opened' THEN 1
        ELSE 0
    END) / SUM(CASE
        WHEN email_category = 'email_sent' THEN 1
        ELSE 0
    END)) * 100 AS open_rate,
    (SUM(CASE
        WHEN email_category = 'email_clickthrough' THEN 1
        ELSE 0
    END) / SUM(CASE

```

```

        WHEN email_category = 'email_sent' THEN 1
        ELSE 0
    END)) * 100 AS click_rate
FROM
    (SELECT
        *,
        CASE
            WHEN action IN ('sent_weekly_digest' , 'sent_reengagement_email')
            THEN ('email_sent')
            WHEN action IN ('email_open') THEN ('email_opened')
            WHEN action IN ('email_clickthrough') THEN ('email_clickthrough')
            END AS email_category
        FROM
            email_events) AS alias;

```

Output :

open_rate	click_rate
33.5834	14.7899

## Insights :

The email engagement metrics show an open rate of approximately 33.58% and a click rate of about 14.79%.

## Result

This project deepened my grasp of operational analytics and data integration. Through SQL analysis, I refined my skills in querying complex datasets. I have learned how to effectively merge and normalize various datasets, ensuring accurate and reliable analysis.

Collaboration with diverse teams honed my communication, teamwork, and ability to translate data insights into actionable recommendations for process improvement.